Package ‘Tendril’

January 20, 2025

Type Package

Title Compute and Display Tendril Plots

License GPL-2

Version 2.0.4

Description Compute the coordinates to produce a tendril plot.
In the tendril plot, each tendril (branch) represents a type of events,
and the direction of the tendril is dictated by on which treatment arm the
event is occurring. If an event is occurring on the first of the two
specified treatment arms, the tendril bends in a clockwise direction.
If an event is occurring on the second of the treatment arms, the
tendril bends in an anti-clockwise direction.
Ref: Karpefors, M and Weatherall, J., ``The Tendril Plot - a novel visual summary
of the incidence, significance and temporal aspects of adverse events in
clinical trials" - JAMIA 2018; 25(8): 1069-1073 <doi:10.1093/jamia/ocy016>.

URL `https://github.com/Karpefors/Tendril`

BugReports `https://github.com/Karpefors/Tendril/issues`

Encoding UTF-8

LazyData true

Depends R (>= 3.5)

Imports ggplot2, dplyr, plyr, reshape2, magrittr, scales, graphics,
grDevices, plotly

RoxygenNote 6.1.1

Suggests testthat, knitr, rmarkdown, devtools

VignetteBuilder knitr

NeedsCompilation no

Author Martin Karpefors [aut],
Stefano Borini [ctb, cre],
Mark Edmondson-Jones [ctb],
Hielke Bijlsma [ctb]

Maintainer Stefano Borini <stefano.borini@astrazeneca.com>

Repository CRAN

Date/Publication 2020-02-11 11:00:02 UTC

Contents

plot.Tendril *Plot Tendril*

Description

Function to plot Tendril results

Usage

```
## S3 method for class 'Tendril'
plot(x, term = NULL, coloring = "Terms",
  interactive = FALSE, ...)
```

Arguments

x	An object of class tendril, as made by Tendril()
term	The term to extract.
coloring	Name of column used for coloring tendrils (only basic plots). Available coloring choices are: "Terms" - One tendril one color; "p" - Prop.test; "p.adj" - P-values adjusted using the False discovery rate method within each tendril; "fish" - Fisher.test; "rdiff" - Risk difference; "RR" - Risk Ratio; "OR" - Odds Ratio; "FDR.tot" - P-values adjusted using the False discovery rate method for all tendrils; and "TermsCount" - Total number of events for that specific type of event
interactive	Specifies if the plot must be interactive or not. If interactive == TRUE, plotly will be used to render the plot. Otherwise, (default) the plot will be rendered as a static image using ggplot2.
...	unused

Details

If saving the results of the function to a variable, this will be of class tendril and will contain the data passed to the plot function and the plot itself

Examples

```
# generate data using Tendril()
data <- Tendril(mydata = TendrilData,
rotations = Rotations,
AEfreqThreshold = 9,
Tag = "Comment",
Treatments = c("placebo", "active"),
Unique.Subject.Identifier = "subjid",
Terms = "ae",
Treat = "treatment",
StartDay = "day",
SubjList = SubjList,
SubjList.subject = "subjid",
SubjList.treatment = "treatment"
)

#Do plot
res <- plot(data, coloring = "Terms")
res <- plot(data, coloring = "p.adj")

#To re-do the plot after the first call:
print(res)
```

`plot.TendrilPerm` *Plot TendrilPerm*

Description

Function to plot TendrilPerm results

Usage

```
## S3 method for class 'TendrilPerm'
plot(x, coloring = "p.adj", percentile = FALSE,
  ...)
```

Arguments

`x` An object of class TendrilPerm, as made by TendrilPerm()

`coloring` Name of column used for coloring tendrils. Available coloring choices are: "Terms" - One tendril one color; "p" - Prop.test; "p.adj" - P-values adjusted using the False discovery rate method within each tendril; "fish" - Fisher.test; "rdiff" - Risk difference; "RR" - Risk Ratio; "OR" - Odds Ratio; "FDR.tot" - P-values adjusted using the False discovery rate method for all tendrils; and "TermsCount" - Total number of events for that specific type of event

`percentile` Specifies if the plot must show percentile values. Default is FALSE.

`...` unused

Examples

```
# generate data using Tendril()
data <- Tendril(mydata = TendrilData,
rotations = Rotations,
AEfreqThreshold = 9,
Tag = "Comment",
Treatments = c("placebo", "active"),
Unique.Subject.Identifier = "subjid",
Terms = "ae",
Treat = "treatment",
StartDay = "day",
SubjList = SubjList,
SubjList.subject = "subjid",
SubjList.treatment = "treatment"
)
tendril_perm <- TendrilPerm(
  tendril = data,
  PermTerm="AE40",
  n.perm = 200,
  perm.from.day = 1)

#Do plot
res <- plot(tendril_perm)
res <- plot(tendril_perm, percentile = TRUE)

#To re-do the plot after the first call:
print(res)
```

`plot_timeseries` *Plot time series*

Description

Plot time series of net events on a second treatment arm

Usage

```
plot_timeseries(tendril, term = NULL)
```

Arguments

`tendril` An object of class Tendril, as made by Tendril()

`term` A character vector describing the value or values of Term to select; defaults to `NULL` which corresponds to all values

Examples

```
# generate data using Tendril()
data <- Tendril(
  mydata = TendrilData,
  rotations = Rotations,
  AEfreqThreshold = 9,
  Tag = "Comment",
  Treatments = c("placebo", "active"),
  Unique.Subject.Identifier = "subjid",
  Terms = "ae",
  Treat = "treatment",
  StartDay = "day",
  SubjList = SubjList,
  SubjList.subject = "subjid",
  SubjList.treatment = "treatment"
)

# do plot
plot_timeseries(data, term="AE33")
plot_timeseries(data, term=c("AE33","AE40"))
plot_timeseries(data, term=NULL)
```

`Rotations` *Example of Rotations in package Tendril.*

Description

Example of Rotations in package Tendril.

Usage

```
Rotations
```

Format

A numeric vector with 1000 number 3s:

`SubjList` *Example of SubjList in package Tendril.*

Description

Example of SubjList in package Tendril.

Usage

```
SubjList
```

Format

A data frame with 500 rows and 2 variables:

subjid : The subject IDs

treatment : The two treatments causing the tendrils to bend left or right

---

Tendril *Tendril*

Description

Function to calculate coordinates and statistical measures used to create a tendril plot

Usage

```
Tendril(mydata, rotations, AEfreqThreshold = 50, Tag = "Comment",
  Treatments = c("Active", "Placebo"),
  Unique.Subject.Identifier = "Unique.Subject.Identifier",
  Terms = "Dictionary.Derived.Term", Treat = "Actual.Treatment...DB",
  StartDay = "Analysis.Start.Relative.Day", SubjList = NULL,
  SubjList.subject = NULL, SubjList.treatment = NULL,
  SubjList.dropoutday = NULL, compensate_imbalance_groups = FALSE,
  filter_double_events = FALSE, suppress_warnings = FALSE)
```

Arguments

`mydata` A dataframe containing the data for the tendril calculations

`rotations` a vector of same length as mydata containing the rotation factors for all the events

`AEfreqThreshold` The minimum frequency threshold of events to be included in the analysis. Default is 50

`Tag` A tag or comment associated with the analysis

`Treatments` The names of the two treatments to be included in the tendril. The first treatment bends to the right and second treatment bends to the left. Must be a vector of two elements and the two elements must be found in the Treatment column

`Unique.Subject.Identifier` The name of the column containing the unique patients IDs

`Terms` The name of the column containing the name of the tendrils (e.g. adverse event terms)

`Treat` The name of the column containing the name of the treatments

`StartDay` The name column containing the days of the events

`SubjList` A dataframe containing subject IDs and treatments

`SubjList.subject` The name of the columns in SubjList containing the subjects IDs

`SubjList.treatment`
: The name of the columns in SubjList containing the treatments

`SubjList.dropoutday`
: The name of the column in SubjList containing the dropoutday

`compensate_imbalance_groups`
: Boolean Whether the rotation factors have been compensated for imbalance in the groups

`filter_double_events`
: Boolean whether to filter out events duplicated in subject id and adverse effect

`suppress_warnings`
: Boolean whether to suppress warnings from chi squared approximation may be incorrect

Details

The function accepts a dataframe with at least 4 columns named as the arguments Unique.Subject.Identifier, Terms, Treat and StartDay.

Two treatments must be given as arguments, and at least one of the two treatments must be found in the Treatment column

The function returns an object of class tendril. The object contains the coordinates for the tendril plot and the arguments of the tendril function

The result of the function can be plotted with plot()

The result can be saved to file using write.table() with argument row.names = FALSE

Value

The function return an object of class tendril. The object contains the original dataset added with the tendril coordinates, all the function arguments and a dataframe with the results from statistical analysis

- data$data : Dataframe of orginal data, coordinates and stat results
- data$Unique.Subject.Identifier : Column containing subject IDs
- data$Terms : Column containing the name of the tendrils
- data$Treat : Column containing the name of the treatments
- data$StartDay : Column containing the days of the events
- data$Treatments : The names of the treatments causing the tendrils to bend
- data$AEfreqThreshold : The frequency threshold for the events to be included in the analysis
- data$Tag : A tag or comment associated with the analysis
- data$n.tot : A dataframe with the total number of events for each treatment. Used in the statistical calculations
- data$SubjList : A dataframe containing subject IDs and treatments
- data$SubjList.subject : The name of the columns in SubjList containing the subjects IDs
- data$SubjList.treatment : The name of the columns in SubjList containing the treatments
- data$SubjList.dropoutday : The name of the column in SubjList containing the dropoutday

- data$rotation_vector : Rotation vector used to generate the tendril
- data$compensate_imbalance_groups : Boolean Whether the rotation factors have been compensated for imbalance in the groups

Examples

```
data <- Tendril(mydata = TendrilData,
rotations = Rotations,
AEfreqThreshold=9,
Tag = "Comment",
Treatments = c("placebo", "active"),
Unique.Subject.Identifier = "subjid",
Terms = "ae",
Treat = "treatment",
StartDay = "day",
SubjList = SubjList,
SubjList.subject = "subjid",
SubjList.treatment = "treatment"
)

plot(data)
```

Tendril.perm.res *Example object Tendril as generated by Tendril() and Tendril.perm().*

Description

The object contains the following 15 items:

Usage

```
Tendril.perm.res
```

Format

A list of 15 elements:

Tendril.perm.res$data : Dataframe of orginal data and coordinates

Tendril.perm.res$Terms : Column containing the name of the tendrils

Tendril.perm.res$Unique.Subject.Identifier : Columns containing subject IDs

Tendril.perm.res$Treat : Column containing the name of the treatments

Tendril.perm.res$StartDay : Column containing the start days of the events

Tendril.perm.res$Treatments : The names of the treatments causing the tendrils to bend

Tendril.perm.res$AEfreqThreshold : The frequency threshold for the events to be included in the analysis

Tendril.perm.res$Tag : A tag or comment associated with the analysis

Tendril.perm.res$SubjList : A dataframe containing subject IDs and treatments

Tendril.perm.res$SubjList.subject : The name of the column in SubjList containing the subjects IDs

Tendril.perm.res$SubjList.treatment : The name of the columns in SubjList containing the treatments

Tendril.perm.res$n.tot : A data frame of 2 variables for number of subjects in each treatment

Tendril.perm.res$PermTerm : Name of the tendril used for the permutations

Tendril.perm.res$perm.data : A dataframe containing the permutation results

Tendril.perm.res$tendril.pi : A list of permutation percentile calculation results

---

`Tendril.res` *Example object Tendril as generated by Tendril().*

Description

The object contains the following 12 items:

Usage

```
Tendril.res
```

Format

A list of 12 elements:

Tendril.res$data : Dataframe of orginal data and coordinates

Tendril.res$Terms : Column containing the name of the tendrils

Tendril.res$Unique.Subject.Identifier : Columns containing subject IDs

Tendril.res$Treat : Column containing the name of the treatments

Tendril.res$StartDay : Column containing the start days of the events

Tendril.res$Treatments : The names of the treatments causing the tendrils to bend

Tendril.res$AEfreqThreshold : The frequency threshold for the events to be included in the analysis

Tendril.res$Tag : A tag or comment associated with the analysis

Tendril.res$SubjList : A dataframe containing subject IDs and treatments

Tendril.res$SubjList.subject : The name of the column in SubjList containing the subjects IDs

Tendril.res$SubjList.treatment : The name of the columns in SubjList containing the treatments

Tendril.res$SubjList.dropoutday : The name of the column in SubjList containing the dropoutday

Tendril.res$rotation_vector : The rotation vector used for the tendril analysis

Tendril.res$compensate_imbalance : Boolean indicating whether the imbalance in the groups is compensated

Tendril.res$n.tot : A data frame of 2 variables for number of subjects in each treatment

TendrilData *Example dataframe in package Tendril.*

Description

A dataset containing the minimal 4 columns required to produe a Tendril plot

Usage

```
TendrilData
```

Format

A data frame with 1000 rows and 4 variables:

subjid The subject IDs

treatment The two treatments causing the tendrils to bend right or left, respectively

ae The names of the tendrils

day The day of the event relative to the treatment start date

TendrilPerm *Tendril permutations*

Description

Function to compute the permutations of one specified tendril, starting from a specific day. Permutations are simulated under the null hypothesis. Thus, on average, there will be an equal number of events on each treatment arm.

Usage

```
TendrilPerm(tendril, PermTerm, n.perm = 100, perm.from.day = 1,
  pi.low = 0.1, pi.high = 0.9)
```

Arguments

`tendril`	an object of class tendril as produced by Tendril()
`PermTerm`	the name of the type of event (tendril) to calculate permutations on
`n.perm`	the number of permutations. Default 100
`perm.from.day`	the starting day for the permutation calculations. Default 1
`pi.low`	percentile low value. Default 0.1
`pi.high`	percentile high value. Default 0.9

Details

Make permutation analysis to a specific type of event, as specified in PermTerm.

Value

The function return an object of class TendrilPerm containing all the input data and a dataframe of permutation results. Use:

data$perm.data

and

data$tendril.pi

and

data$tendril

to access the permutations, percentiles dataframes, and tendril data respectively

Examples

```
# Create tendril
tendril <- Tendril(mydata = TendrilData,
  rotations = Rotations,
  AEfreqThreshold = 9,
  Tag = "Comment",
  Treatments = c("placebo", "active"),
  Unique.Subject.Identifier = "subjid",
  Terms = "ae",
  Treat = "treatment",
  StartDay = "day",
  SubjList = SubjList,
  SubjList.subject = "subjid",
  SubjList.treatment = "treatment"
)

# Compute permutations
perm.data <- TendrilPerm(tendril = tendril,
  PermTerm="AE40",
  n.perm = 200,
  perm.from.day = 1)

# Plot results
plot(perm.data)
plot(perm.data, percentile = TRUE)
```

Index